invention as claimed in the Patent Application and the prima facie patentability of each of the claims recited therein, notwithstanding the assertions of the Examiner in paragraph 1 of the Office Action.

The Applicants have invented a method and apparatus for high speed interprocess communications ("IPC"). Conventionally, multiple processes can communicate via the use of a shared region of random access memory ("RAM") to which each process can write data, and from which each process can read data. When communicating through the shared region of RAM, a first process functioning as the message source can write the message to the shared region of RAM. Subsequently, the second process, functioning as the message receiver, can read the written message from the shared region of RAM. Thus, minimally, two system calls are required to move n bytes of data from the first process to the second process through the shared region of RAM. Moreover, 2*n bytes of data will be stored in total—n bytes into the shared region of RAM, and n bytes into user memory space associated with the second process.

To overcome the excessive overhead associated with conventional IPC utilizing a shared region of RAM, the high speed IPC method and apparatus of the present invention avoids moving 2*n bytes of data by passing to the second process, not a full copy of the data, but merely a memory offset into the shared region of RAM from which point the second process can access the data. Advantageously, the memory offset can be absolute relative to a commonly known address in the shared region of RAM so as to avoid problems arising from different memory mappings by each process of the same shared region of RAM. In this way, message passing in the high speed IPC method and apparatus of the Applicants' invention does not require storing message data in operating system kernel space. As such, system calls further are not required to write and read data. Thus, the elevated risk associated with using operating system kernel space arising from the loss of CPU control by a communicating process can be eliminated.

Claims 1 and 13 of the Patent Application can be graphed as follows:

A method or computer apparatus for high speed IPC having the following steps:

1. Attaching first and second processes
   A. to a message buffer

i. in a shared region of RAM
a. exclusive of operating system kernel space
B. each process having a message list.

2. Accumulating message data
A. from said first process
B. in a location in said message buffer.

3. Adding to said message list of said second process
A. a memory offset
i. corresponding to said location in said message buffer.

4. Processing in said second process
A. said accumulated data
i. at said location corresponding to said offset.

Importantly, one skilled in the art will recognize the pertinence to the Applicants' invention of step 3.A.i, namely adding to a message list of a second process, a memory offset which corresponds to a location in the message buffer of the accumulated message data from the first process.

Turning now to the rejections on the art, in paragraph 1 of the Office Action, claims 1 through 18 have been rejected as an obvious variation of the combination of Haba and Carter. Carter relates to the creation and management of virtual memory space that can be shared by each computer on a network, and which further can span the storage space of each memory device connected to the network. Seemingly, the Examiner has relied upon the Carter reference for the proposition that a shared region of RAM exclusive of operating system kernel space might be used during the course of IPC. Indeed, Carter teaches "virtual shared memory" which can be accessed by a shared memory subsystem which is not included as part of the operating system kernel space. Carter, however, precisely illustrates the deficiencies of the prior art addressed directly both by the Applicants' invention and claims directed thereto.

Specifically, to transfer data from one node to another using the shared region of RAM in Carter, first, n bytes of data can be written by the source node to the shared region of RAM. Subsequently, the n bytes of data can be read out to the receiving node. Of course, in addition to moving 2*n bytes of data in the Carter system, to undertake the write and read operations, minimally two system calls will be required. Nevertheless, as Carter teaches a complex method of managing shared regions of

RAM, additional system calls will be required such as calls to the "global RAM directory" to locate the precise region of shared RAM from which the data can be located. In this regard, Carter teaches away from the Applicants' invention which is directed toward a reduction in processing overhead when undertaking IPC—not an enlargement of processing overhead. Thus, it would not be appropriate to combine Carter with any reference in support of a rejection of the claims of the Patent Application.

Notwithstanding the citation of Carter, the Examiner primarily has relied upon Haba in support of the obviousness rejection of claims 1 through 18. Haba relates not to high speed IPC, but to a more efficient means for flushing data to a mass storage device. In particular, in Haba it has been recognized that "hardware devices", which are defined to be hard disk drives, and more particularly, which are not to be defined as RAM, are the source of performance bottlenecks in as much as the operative rate of a hardware device pales in comparison to the performance of RAM. Yet, in Haba it is further recognized that RAM cannot provide the persistence necessary to address the reliability concerns of many applications.

To address the performance bottleneck of conventional disk drives, in Haba it is proposed that the accumulation of data to be written to disk occur concurrently in a separate programming thread to the flushing of accumulated data to the disk drive. Specifically, as recited in column 2, lines 14 through 24:

> To overcome the processing limitations of the traditional methods, the performance of a write operation is divided between a data collector and a data writer which operate in different threads...The data writer manages the writing of data elements to the hardware device, while the data collector accumulates, orders and consolidates the data items for efficient storage by the data writer.

Lines 25 through 38 continue:

> Any data items to be written to disk are passed to the data collector from a data source...Because the data storage rate of the hardware device is relatively slow compared to a write to RAM especially when random portions of a hardware device are accessed, the present invention employs a method to keep the data writer storing data to the disk...Moreover, because writing larger blocks of contiguous data items is more efficient than a plurality of write operations on individual contiguous data items, the data

collector consolidates accumulated multiple write request of contiguous data items into a single write request.

To effectuate the writing of larger blocks of contiguous data items, as discussed in column 7, lines 17-30 of the Haba specification, a data collector can receive a stream of write requests, both accumulating and consolidating the received requests in a data structure. A data writer, operating in parallel and in a separate thread than the data collector, can store the previously acquired write requests to the disk drive. When the data writer has no more data items to store, the data writer can acquire the newly received accumulated and consolidated write requests from the data collector. In this manner, the data writer constantly stores data items to disk while the data collector constantly and concurrently receives, accumulates and consolidates requests to store data items to disk.

A comparison of the Haba technology to the high speed IPC process and apparatus of the Applicants' invention will reveal that Haba inherently is not compatible with the claimed intent and implementation of the Applicants' high speed IPC process. Most apparently, whereas the high speed IPC process of the Applicants' invention aims to reduce resource overhead by avoiding slower speed data transfers, Haba explicitly selects slower speed hardware devices over high speed RAM. Haba actually acknowledges the "slowness" of disk technology over RAM. It should further be recognized that Haba does not teach any IPC techniques. Rather, Haba seeks to improve disk writing operations exclusively. Communications between two processes is not of concern in the Haba technology. In consequence, one seeking to solve problems associated with IPC techniques would not turn to Haba and other disk writing technologies for solutions. Thus, it would not be legally appropriate to combine Haba with Carter in support of an obviousness rejection of a claim directed to high speed IPC.

In any event, the combination of Haba and Carter wholly fails to teach the Applicants' invention as recited in claims 1 through 18. In this regard, by reference to the claim graph textually illustrated *infra* on page 3, to support the rejection of claim 1 or claim 13, it will be required that either Haba, Carter or the combination of Haba and Carter teach the attachment of two processes to a message buffer in a shared region of RAM. While Haba does not teach the attachment of any processes to a message buffer in RAM, neither can it be said that Carter teaches the attachment of two processes to a

message buffer in RAM. Rather, in Carter processes attach to only to one of a memory controller, directory service and flow scheduler. Thus, step A is lacking in both Carter and Haba.

Secondly, step D requires that a second process, process the accumulated data at the location corresponding to the offset. In Carter, data is transferred from the shared memory to local memory as would have been the case in conventional shared memory configurations. In Haba, by comparison, data is transferred from one queue to another where it is processed locally by the data writer when the data is written to disk. Accordingly, it cannot be said that Haba process the accumulated data at the location corresponding to the offset.

Finally, and most importantly, nowhere in Haba and in Carter, has it been shown that a memory offset is added to the message list of the second process. In Carter, data is passed by value and not by reference from process to process. In Haba, too, data is passed by value as it is flushed to disk. Indeed, page 8, lines 18-19 of the Patent Application explicitly state, "Processes are notified of the location for the message data rather than actually receiving a copy of the message data." So much has been recited in claims 1 and 13 by reference to step C. Accordingly, as the combination of Haba and Carter lack any recitation to steps A, C and D, all of the claims recited in the Patent Application are prima facie allowable over the cited references.

In conclusion, the Applicants respectfully request the withdrawal of the rejections under 35 U.S.C. § 103(a) based upon the Applicant's foregoing remarks. In that regard, each of claims 1 through 18 are believed to be allowable, and accordingly, the entire Patent Application is believed to be in condition for allowance. Consequently, such action is respectfully requested. To that end, the Applicants request that the Examiner call the undersigned if clarification is needed on any matter within this Amendment, or if

the Examiner believes a telephone interview would expedite the prosecution of the subject application to completion.

Respectfully submitted,

Date: 1/8/03

Robert J. Sacco, Registration No. 35,667
Steven M. Greenberg, Registration No. 44,725
AKERMAN SENTERFITT
222 Lakeview Avenue, Suite 400
Post Office Box 3188
West Palm Beach, FL 33402-3188
Telephone: (561) 659-5000

Docket No. 6572-14